Inserting Software Fault Measurement Techniques into Development Efforts

Allen P. Nikora
Jet Propulsion Laboratory, California Institute of Technology
E-mail: Allen.P.Nikora@jpl.nasa.gov

Abstract

Over the past several years, techniques for estimating software fault content based on measurements of a system's structural evolution during its implementation have been developed. Proper application of the techniques will yield a detailed map of the faults that have been inserted into the system. This information can be used by development organizations to better control the number of residual faults in the operational system.

There are several issues that must be resolved if these techniques are to be successfully inserted into a development effort. These issues are identified, as are possibilities for their resolution.

1. Introduction

As our reliance on software-controlled systems increases, it becomes increasingly important that they behave reliably during fielded use. To develop reliable software, it is essential to develop accurate and practical methods of monitoring its reliability during development. Approximately 30 years ago, the first software reliability models were published [1, 2]. Additional models and methods for increasing predictive accuracy have been developed over the intervening years [3]. These types of models allow reliability estimates and forecasts to be made during the testing phases – development managers are able to use these models to help answer the following questions:

1. Does the software now meet the required reliability?
2. How much more time and effort will be required to achieve the required reliability?
3. If insufficient resources are available, what will the reliability of the fielded system be?

Although these types of models can be useful in managing testing resources, they can be used only after the system has already been implemented. This means that the results of these models cannot be used to guide the system's implementation and design so as to optimize its quality. If methods of estimating a system's quality based on structural characteristics or characteristics of the development environment could be developed, they might be used to more effectively manage quality during early phases of a development effort.

2. Fault Measurement and Estimation

One method of estimating reliability-related quantities uses measurements of its structural evolution during development as a surrogate for the system's fault content [4, 5]. Briefly, the structural change between subsequent revisions of each module in the system is measured with respect to a chosen baseline. For each module, this set of measurements is proportional to the fault burden of that module. For example, if a module has received 7% of the total measured change made to the system during its development, we can expect that 7% of the total number of faults inserted into the system will have been inserted into that module. This information can be used during implementation to identify those modules in which the greatest number of faults can be expected. Additional fault identification resources (e.g., inspectors, testers) can be allocated to these modules.

If it is possible to identify the revision of a module into which a discovered fault was inserted, a regression model relating the amount of structural change to the number of faults inserted into the system can be developed. If a development effort's problem reporting system tracks the underlying faults and the repairs made to the affected modules, the residual number of faults in each module can also be estimated.

3. Practical Measurement Issues

In implementing the measurement techniques described above, the following issues arise:

- Measurements across a single development effort, or among multiple efforts, must be comparable. To identify those areas within a single development effort having the highest fault rates, it must be possible to compare measurements across all areas of that effort. When using the techniques to measure an organization's progress in developing more reliable software, it must be possible to compare measurements of several development efforts. To make this possible, we have developed standards for:

- Structural measurements.
- Fault identification and counting.

- Measurement must be performed in a well-defined and consistent fashion. For many development efforts, a policy for measuring the system's structural change during development can easily be implemented as part of the configuration management infrastructure. As developers check new revisions of software components into the CM library, the structural measurements and analyses are performed as part of the check-in process, thereby enforcing the policy. A pilot effort at JPL to implement the measurement techniques described in [4, 5] in this fashion is currently in progress.

  Consistently measuring faults is conceptually straightforward. For each failure recorded in the problem tracking system, the fault identification rules are applied to the changes made to the system in response to the failure. To accurately count faults, all failures must be recorded in the problem reporting system. More importantly, changes made to a component in response to a failure must be distinguishable from those made to modify or add functionality. This last can be facilitated to some extent by having the problem reporting system integrated with the configuration management system – each time a failure is reported, a new change package with a name identifying it with a particular failure report is automatically created. If developers restrict themselves to only checking repairs to the fault components into the change package, changes due to fault repair can readily be distinguished from other types of changes. We are currently evaluating the effectiveness of this scheme for the JPL pilot effort.

  Application of the fault identification rules to the changes made in response to a failure is, at this point, a largely manual process. The resolution of this issue is beyond the scope of the JPL pilot effort. In the future, we hope to devise methods of applying some degree of automation to this aspect of measurement. If the fault identification rules we have devised can be formalized, it may be possible to develop tools that can be used to assist the search for the initial instance of a given fault. This would make it easier to obtain the measurements needed to estimate a component's absolute fault burden.
- Developers and managers must be able to use the measurements to help them better perform their tasks. To accomplish this, measurements must relate to properties of the software that developers and managers want to control, and measurements and analysis results must be expressed in a readily understandable form. The techniques being implemented in the JPL pilot effort produce measurements that are strongly related to a component's fault content, so the querying mechanism used by developers and managers will report results as estimated proportional or absolute fault burdens at varying levels of detail. We are currently working with developers and managers at JPL to develop appropriate methods of retrieving and presenting the measurement results.

4. Acknowledgments

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology. Portions of the work were sponsored by the National Aeronautics and Space Administration's IV&V Facility

5. References

[1] Z. Jelinski, P. Moranda; "Software Reliability Research," Proceedings of the Statistical Methods for the Evaluation of Computer System Performance, Academic Press, 1972, pp. 465-484

[2] M. L. Shooman, "Probabilistic Models for Software Reliability Predictions," Statistical Computer Performance Evaluation, Academic Press, New York, June, 1972, pp. 485-502

[3] M. Lyu ed., Handbook of Software Reliability Engineering, McGraw-Hill, 1996, ISBN 0-07-039400-8, pp.493-504

[4] J. Munson, A. Nikora, "Estimating Rates of Fault Insertion and Test Effectiveness in Software Systems", proceedings of the Fourth ISSAT International Conference on Quality and Reliability in Design, Seattle, WA, August 12-14, 1998

[5] A. Nikora, J. Munson, "Determining Fault Insertion Rates For Evolving Software Systems", proceedings of the Ninth International Symposium on Software Reliability Engineering, Paderborn, Germany, November 4-7, 1998